

# *Multi-attribute collaborative filtering recommendation based on improved group decision-making*

Article

Published Version

Open access

Yu, C., Yan, L. and Liu, K. (2015) Multi-attribute collaborative filtering recommendation based on improved group decision-making. *International Journal of Computers Communications & Control*, 10 (5). pp. 746-759. ISSN 1841-9836 doi: <https://doi.org/10.15837/ijccc.2015.5.1379> Available at <https://centaur.reading.ac.uk/42567/>

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

Published version at: <http://dx.doi.org/10.15837/ijccc.2015.5.1379>

Identification Number/DOI: <https://doi.org/10.15837/ijccc.2015.5.1379>  
<<https://doi.org/10.15837/ijccc.2015.5.1379>>

Publisher: IJCCC

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the [End User Agreement](#).

[www.reading.ac.uk/centaur](http://www.reading.ac.uk/centaur)

**CentAUR**

Central Archive at the University of Reading

Reading's research outputs online

# Multi-attribute Collaborative Filtering Recommendation

C. Yu, Y. Luo, K. Liu

## Changrui Yu

School of Information Management and Engineering,  
Shanghai University of Finance and Economics,  
Shanghai 200433, China  
yucr@sjtu.edu.cn

## Yan Luo\*

SHU-UTS SILC Business School,  
Shanghai University,  
20 Chengzhong RD, Shanghai 201800, China  
\*Corresponding author: luoyan@shu.edu.cn

## Kecheng Liu

Informatics Research Centre, Henley Business School,  
University of Reading,  
Reading, RG6 3XA, U.K.  
k.liu@reading.ac.uk

**Abstract:** Currently researchers in the field of personalized recommendations bear little consideration on users' interest differences in resource attributes although resource attribute is usually one of the most important factors in determining user preferences. To solve this problem, the paper builds an evaluation model of user interest based on resource multi-attributes, proposes a modified Pearson-Compatibility multi-attribute group decision-making algorithm, and introduces an algorithm to solve the recommendation problem of  $k$ -neighbor similar users. This study addresses the issues on preference differences of similar users, incomplete values, and advanced converge of the algorithm, and realizes multi-attribute collaborative filtering. The algorithm is proved to be effective by an experiment of collaborative recommendation among multi-users in a virtual environment. The experimental results show that the algorithm has a high accuracy on predicting target users' attribute preferences and has a strong anti-interference ability on deviation and incomplete values.

**Keywords:** personalized recommendation, group decision-making, multi-attribute, collaborative filtering, Pearson-Compatibility.

## 1 Introduction

A recommender system aims to generate meaningful recommendations to users for items or products that might interest them [1]. In many markets, consumers face a wealth of products and information from which they can make choices. To alleviate this problem, many web sites attempt to help users by incorporating a recommender system that provides users with a list of items and/or web pages that are likely to interest them. There are real-world operations of industrial recommender systems, such as the recommendations for books on Amazon, or movies on Netflix.

As one of the most successful approaches to building recommender systems, collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users [2]. The developers of one of the first recommender systems, Tapestry [3] coined the phrase "collaborative filtering (CF)", which has been widely adopted regardless of the facts that recommenders may not explicitly collaborate

with recipients and recommendations may suggest particularly interesting items, in addition to indicating those that should be filtered out [4]. The fundamental assumption of CF is that if users  $X$  and  $Y$  rate  $n$  items similarly, or have similar behaviors (e.g., buying, watching, and listening), and hence will rate or act on other items similarly [5].

Studies in literature [3] and [4] have shown that users' interest in a product or service is affected by user topic preferences, content preferences, user habits, public evaluation and other factors, and that these factors are decided by the different attributes of items. For example, users liking a new movie may be caused by one or more attributes of the movie, such as the director, star, theme, content, style, public comment and so forth. Thus, in the application of collaborative filtering algorithm, it is necessary to use a multi-attribute analysis model, in which the user rating to an item is based on a different perspective (attributes) to describe their interest preferences.

Although user and resource clustering based on the attribute information has been widely discussed in the multi-attribute collaborative filtering literatures, the recommend method is still traditional [6]. Such methods can only obtain a set of potential interest items of target users, but the reasons of such a recommendation are not given to the target user. In addition, the present studies scarcely consider the characteristics differences of similar users interested in the item attributes, which can lead to recommendation deviation [6]. For example, in a traditional way, user  $B$  is the most similar one to the target user  $A$  because  $A$  and  $B$  have the same degree of interest in the same film. However, if the film properties they prefer are completely different, it will lead to recommendation deviation when we give greater weight to  $B$ .

Based on our previous research, we propose that multi-attribute collaborative filtering can be treated as a group decision making process. By building the rating matrix of target items for the similar users, we remove the user who has a large attribute preference difference to target user from the nearest user set, and save the problem of recommendations deviation. In addition, we can analyze the user's interest performance from the view of item attributes and give the descriptions for the recommendation. Accordingly, this paper proposes a modified Pearson-Compatibility multi-attribute group decision-making algorithm and introduces the algorithm to solve the recommendation problem of  $k$ -neighbor similar users.

This paper has extended our previous research significantly with much more details of the theoretical model, such as the characters of the multi-attribute evaluation, the selection of the nearest neighbor of target user, and the group decision-making model of personalized recommendation. More importantly, we have enriched and refined the collaborative filtering recommendation algorithm which is the core of this paper.

This paper is organized as follows. We review recommender systems and multi-attribute utility theory. We then introduce the establishment of user interest model using applied ontology method to describe user profile and illustrate the algorithm in specific steps. Finally, we do an experiment and conclude with findings and discussions.

## 2 Descriptions of Basic Model

### 2.1 User rating matrix

A user's comment on a certain item is usually an integration of multi-attribute comments made from different angles [6]. Suppose an item is shown as follows:

$$P = \{a_1, a_2, a_3, \dots, a_n\}$$

Based on the revised rating model, the paper establishes the user rating matrix. Suppose the user set is denoted as  $U = \{U_1, U_2, \dots, U_p\}$  and the user  $U_j$  rating for item  $P_i$  is denoted as

$A(U_j, P_i)$ :

$$A(U_j, P_i) = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_{n-1} & a_n \\ \omega_{11} & \omega_{12} & \omega_{13} & \dots & \omega_{1n-1} & \omega_{1n} \\ \omega_{21} & \omega_{22} & \omega_{23} & \dots & \omega_{2n-1} & \omega_{2n} \\ \omega_{31} & \omega_{32} & \omega_{33} & \dots & \omega_{3n-1} & \omega_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega_{(n-1)1} & \omega_{(n-1)2} & \omega_{(n-1)3} & \dots & \omega_{(n-1)(n-1)} & \omega_{(n-1)n} \\ \omega_{n1} & \omega_{n2} & \omega_{n3} & \dots & \omega_{n(n-1)} & \omega_{nn} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_{n-1} \\ a_n \end{bmatrix}$$

Where  $\omega_{xy}$  is the importance of attribute  $a_x$  of product  $P_i$  in comparison with attribute  $a_y$  for user  $U_j$ . Here we use the 1-9 scale Paired comparison method to analyze the compared importance level of each attribute of the product that a user evaluates [7]. The rating matrix of an item is mainly acquired through user scoring, or acquired through user behavior analysis, or acquired with the approaches of Web semantic digging and fuzzy mathematics [6].

The present user rating system, such as MovieLens, only asks a user to make a synthetic rating for the movie he or she watched and give a quantitative scoring between 1 and 5. This approach is not accurate in identifying the similarity in the preference of two users. For example, if two users are interested in the same movie. When they rate it, they give it the same score. However, the angles of their preference for the movie are totally different. A user may like the star and the style. The other user prefers the theme and the content of the movie. Therefore, we propose to build a multi-attribute rating system to evaluate a product, i.e., a product has many attributes. When evaluating the product, a user mainly gives the preference ratings of each product attribute.

Another notable character of the multi-attribute evaluation is the absolute sparsity of the judgment matrix. This is caused by users who only know a few attributes of the product. This matrix is like the incomplete value judgment matrix of group-decision making, which easily lead to obvious deviations for the prediction results. So we should take some adjusted measures to solve the problem.

## 2.2 User Interest Model

Suppose user  $U_j$  has rated several items and the rating matrix set is  $AS = \{A(U_j, P_1), A(U_j, P_2), \dots, A(U_j, P_t)\}$ , where  $A(U_j, P_i) (i = 1, 2, \dots, t)$  is user  $U_j$  rating matrix for product  $i$  (i.e.,  $P_i$ ). This paper applies the rating matrix set to establishing the user interest model [6]. The specific steps are as follows.

1. Calculating the feature weight vector of each rating matrix, and then acquire the feature weight vector set

$$VS = \{V_{U_j}^{P_1}(w_1, w_2, w_3, \dots, w_{size(A(U_j, P_1))}), V_{U_j}^{P_2}(w_1, w_2, w_3, \dots, w_{size(A(U_j, P_2))}) \dots V_{U_j}^{P_t}(w_1, w_2, w_3, \dots, w_{size(A(U_j, P_t))})\}.$$

Where  $V_{U_j}^{P_i}(w_1, w_2, w_3, \dots, w_{size(A(U_j, P_i))})$  denotes the feature weight vector of the user rating matrix  $A(U_j, P_i) (i = 1, 2, \dots, t)$  and  $size(A(U_j, P_i))$  denotes the length of the feature weight vector.

2. According to the category of each attribute, calculate the user interest weights of the relevant attribute in the related resource category. Referring to the methods proposed

by [8] and [9], we propose the following formula for calculating the degree of the user interest.

$$Va(U_j, a_y, n) = \frac{\sum_{k=1}^n A(U_j, P_k) \times V_{U_j}^{P_k}(w_y)}{n} \quad (1)$$

Where  $Va(U_j, a_y, n)$  denotes the degree to which user  $U_j$  is interested in attribute  $a_y$ .  $n$  is the number of the items which has attribute  $a_y$  and user  $U_j$  have rated.  $A(U_j, P_k) \times V_{U_j}^{P_k}(w_y)$  ( $k = 1, 2, 3, \dots, n$ ) denotes the degree of user  $U_j$  interest in attribute  $a_y$  of product  $P_k$ , which indicates how user  $U_j$  preference on item  $P_k$  is mostly determined by attribute  $a_y$ .

### 2.3 Selection of the Nearest Neighbor of Target User

We define the target user as the online user which requires evaluations and preliminary recommendations. The set of nearest neighbors is composed of the users who have the most similar interest and preference to item with the target user.

Traditionally, researchers use  $k$ -Nearest Neighbor (KNN) algorithm and Pearson correlation-based similarity formula to do cluster analysis on the target user and the similar users, according to their similar interest and preference. Through these methods, the similar user set with different group standards can be obtained. In the process of collaborative filtering recommendation with group decision-making method, we consider the characteristics of the target user preference and search the similar user sets. The specific steps are as follows.

Step 1. Search the user set which has similar interest distribution with target user  $U_T$ , i.e., to obtain the intersection set of the interest distribution of all users and the target user. Two users can rate the same attributes in a number of categories and have similar interest weights. This approach can be applied to obtain the initial Nearest Neighbor set:

$$IU = \{(U_1, \Omega_1), (U_2, \Omega_2), (U_3, \Omega_3) \dots, (U_w, \Omega_w)\}$$

Where  $w$  is the number of users totally in the initial Nearest Neighbor set.  $u_k$  denotes the  $k^{\text{th}}$  user whose interest set is  $S_k$ .  $\Omega_k$  denotes the interest intersection between user  $U_k$  and the target user  $U_T$ .

Step 2. Use Pearson correlation-based similarity formula to calculate the similarity between target user  $U_T$  and a random user  $U_k$ . When the degree of similarity reaches the threshold,  $u_k$  can be divided into  $S_{U_T}$  which is the similar user set of  $U_T$ . Finally,  $S_{u_T} = \{u_1, u_2, \dots, u_s\}$  is obtained, i.e.,  $s$  similar users meet the threshold. The set of the interest intersection between the target user  $u_T$  and  $s$  similar users that meet the threshold is  $Se_{u_T} = \Omega_1 \cup \Omega_2 \cup \Omega_3 \dots \cup \Omega_s$ .

## 3 Collaborative Filtering Recommendation Algorithm Based on Multi-attribute Group Decision-making

### 3.1 Group Decision-making Model of Personalized Recommendation

After acquiring a similar user set  $S_{U_T}$ , we need predict and recommend the items that target user  $U_T$  has not commented yet. Suppose the item set is  $Source = \{P_1, P_2, P_3, \dots, P_n\}$  and  $P_k$  ( $k = 1, 2, \dots, n$ ) is the item that the target user has not commented yet. The traditional collaborative filtering recommendation algorithm is applied to calculate the overall evaluation value of the item  $A(U_T, P_k)$  given by similar users and then obtain the initial recommended source  $S_{Initial}$ . Suppose a random item  $P_k \in S_{Initial}$  has  $n$  attributes denoted as a set  $Sp_k =$

$\{a_1, a_2, \dots, a_n\}$ , the set satisfies the condition  $Sp_k \subseteq Se_{u_T}$ . The comment matrix is denoted as  $A^1, A^2, A^3, \dots, A^p$ , which means similar users in  $S_{u_T}$  commend  $P_k$ . We suppose  $p$  users are similar with the target user and have made comments to the item. The algorithm proposed in this paper requires that  $p \geq 3$ . When  $p < 3$ , refer to the article [10].

After obtaining the evaluation matrix  $A^1, A^2, A^3, \dots, A^p$ , it is vital for this research to use the information of comments to get preference matrix of the target users. The collaborative filtering recommendation of multi-attribute similar users is a typical group decision problem, in which many similar users make their decisions independently without any discussions and then the computer synthesizes their opinions to make recommendations to the target user. The result requires that all similar users' comprehensive evaluation matrixes and comprehensive characteristics weight vector  $\{w_1, w_2, w_3, \dots, w_n\}$  be calculated.  $w_i$  denotes the comprehensive preference of the similar users to attribute  $a_i$ , which is compared with other attributes of the product that has not been recommended. According to comprehensive characteristics weight vector, we can know which attribute determines the users' interest on a specific item. Applying the semantic analysis method and the semi-structured description language to the attributes, we can make better recommendation to users.

A recommendation model can be transferred to group decision-making model in order to solve a problem. However, the traditional group decision-making algorithm still needs to be improved when applied since there are differences between similar users and decision-making experts. The differences mainly exist in the following aspects:

1. It is hard to use the weight vectors to measure the influence of similar users in the recommendation processes mainly because of the complexity of user preferences. According to the traditional method, two users have a high similarity in their interest preferences. However, it is hard to identify the deviation existing in the preferences of the two users on a specific item.
2. There are a large number of incomplete values in user comment information because some users may make no comments on the unfamiliar attributes to ensure the accuracy of evaluations. In this situation, the application of some traditional group decision-making algorithms, such as weighted average method or weighted least squared logarithm method may result in a great deviation of the final result.

To solve the two problems mentioned above, this paper makes improvements in group decision-making compatibility test algorithm and builds a collaborative correction algorithm based on Pearson-Compatibility model. The core of this algorithm is to simulate the expert group-decision making process, discover the common opinions of most experts via data analysis, and revise the opinions of each expert to realize the final consistent compatibility. Moreover, to solve the problem of premature convergence, this paper adopts a mode of multi-user collaborative correction. In addition to the usual algorithms, when it calculates the value of each user's impact, this paper uses the common opinion of all users rather than target user's opinion as a measurable index. The user whose opinion is more similar to the common opinion would be given a higher weight, which is different to traditional method.

### 3.2 The Related Definitions

Firstly, we introduce the calculation of the value of user impact weight. This value is an important indicator to measure the degree of evaluation information consistency between a user and the others [6]. The user matrix with higher group evaluation consistency will get higher

weight. Vice versa. This paper adopts the concept of user rating similarity [11, 12]. We turn all the similar  $n \times n$  user rating matrixes into  $n^2 \times 1$  one dimensional vector. The user  $U_k$  judgment matrix  $A^k$  could be denoted as  $V^k = \{\omega_{11}^k, \omega_{12}^k, \omega_{13}^k \dots \omega_{1n}^k, \omega_{21}^k, \omega_{22}^k, \omega_{23}^k \dots \omega_{2n}^k \dots \omega_{n1}^k \dots \omega_{nn}^k\}$ . Pearson similarity formula to calculate the rating matrix between the user  $U_k$  and the user  $U_l$  is show as follows:

$$Si(A^k, A^l) = \frac{\sum_{i=1}^{n^2} (V^k(i) - \overline{V^k}) \times (V^l(i) - \overline{V^l})}{\sqrt{\sum_{u=1}^{n^2} [V^k(i) - \overline{V^k}]^2} \times \sqrt{\sum_{u=1}^{n^2} [V^l(i) - \overline{V^l}]^2}} \quad (2)$$

$\overline{V^k}$  is the average value of all elements of user  $U_k$  rating matrix.

$$\overline{V^k} = \frac{\omega_{11}^k + \omega_{12}^k + \dots + \omega_{nn}^k}{n^2}.$$

The similarity between user  $k$  and other users could be calculated as follows:

$$Si_k = \sum_{l=1, l \neq k}^p Si(A^k, A^l) / (p - 1).$$

where  $p$  denotes the number of users. We propose a formula  $D_k = 1 - Si_k$  as the approximate measure of variance, which indicates the deviation degree of evaluation matrix. The approximate influence weight of user  $U_k$  is shown as follows:

$$\theta_k = \frac{(1 - \max\{Si_l, l = 1, 2, \dots, p\})^2}{D_k^2} \quad (3)$$

After acquiring the similar user influence weight, we suppose the group integrated approximate evaluation matrix of  $p$  users is  $A^*$ , and the value of each element  $\omega_{ij}^*$  in matrix  $A^*$  is as following:

$$\omega_{ij}^* = \sum_{k=1}^p \theta_k \times \omega_{ij}^k / \sum_{k=1}^p \theta_k \quad (4)$$

Matrix  $A^*$  is not a positive reciprocal matrix. Suppose  $X$  is a positive reciprocal matrix composed of  $x_{ij}$ . This paper uses the least square method to modify  $X$  and propose the following formula:

$$F(X) = \min \sum_{i=1}^n \sum_{j=1}^n (x_{ij} - \omega_{ij}^*)^2$$

$$s.t \begin{cases} x_{ij} \times x_{ji} = 1 \\ x_{ij} > 0 \end{cases} \quad (i, j = 1, 2, \dots, n) \quad (5)$$

The definition of compatibility and comprehensive compatibility are as follows:

**Definition 1.** Suppose  $X$  is the group user comprehensive evaluation matrix obtained by using the method of the least squares. Then the judgment matrix compatibility between user  $k$  and the other users is as follows:

$$S(A^k, X) = \frac{\sum_{i=1}^n \sum_{j=1}^n \frac{\omega_{ij}^{(k)} \times x_{ij}}{\max((\omega_{ij}^{(k)})^2, (x_{ij})^2)}}{n^2 - \alpha} \quad (6)$$



Although paper [13] has defined expert judgment matrix compatibility in usual cases, it does not consider the incomplete value. Formula (6) is a modified approach to solve the problem. Firstly, the block that the user does not rate is processed and given the value 0. Then  $\alpha$  is used to indicate the number of 0. The aim of this approach is to eliminate the influence of user judgment matrix on compatibility indicator.

**Definition 2.** Suppose  $A^{1'}, A^{2'} \dots A^{p'}$  are the compatibility correction of matrixes of  $p$  users' judgment matrixes. Then we get the comprehensive consistency indicator  $\bar{S}$ , as follows:

$$\bar{S} = \frac{\sum_{k=1}^p S(A, A^{k'})}{p} \quad (7)$$

Readers can refer to the simulation result of article [7]. When  $S(A, B) \geq 0.8$ , the two evaluation matrixes is considered nearly compatible. When  $\bar{S} \geq 0.8$ , evaluation matrixes of all  $p$  similar users is considered compatible.

### 3.3 Collaborative Correction Algorithm Based On Pearson-Compatibility

Pearson-Compatibility model could be used to simulate the process of experts doing group discussions and finally making group decisions. The model mainly consists of two indicator calculation formulae: Pearson similarity calculation and Compatibility test. Pearson similarity calculation formula is mainly used to calculate the information reservation degree of user rating matrix after the matrix has been revised. And Compatibility test is used to test degree of consistency of user rating matrix after matrixes has been revised. The mainly function of Pearson-Compatibility model is to build the associations between two variational indicators. Based on this association, Compatibility correction algorithm will choose the best revised matrix in each iteration process, which could make the user matrixes unanimous and most similar to the true value.

Referring to [14], based on the related method [15, 16], this paper proposes a Pearson-Compatibility model as follows:

$$\Omega^{(t)} = \overline{S_i^{(t)}}^{1-\beta} \times \overline{S^{(t)}}^{1+\beta} \quad (8)$$

Where  $\Omega^{(t)}$  denotes the calculation result of Pearson-Compatibility model after  $t$  times iteration. With all the continuous revision of user matrixes,  $\overline{S_i^{(t)}}$  decreases from 1 to 0, and  $\overline{S^{(t)}}$  increases from 0 to 1. So each time of modification generates a better revised scheme which minimizes the change of  $\overline{S_i^{(t)}}$  while maximizes the change of  $\overline{S^{(t)}}$ . This indicates that all the experts acquires a better compatibility with the least possible original information lose. This method is similar to the real decision processes [16] and can provide an effective way to approach the true value. In Pearson-Compatibility model, the best revised scheme in each iteration process is the one that maximizes  $\Omega^{(t)}$ .  $\beta$  is used to control the marginal diminishing rate and the marginal increasing rate of  $\overline{S_i^{(t)}}$  and  $\overline{S^{(t)}}$ . The efficiency of this algorithm is low when the consistency of the evaluation matrixes is low. Moreover, it is possible that the algorithm converges in advance, i.e.,  $\Omega^{(t)}$  reaches the max value at the beginning of calculation. At that time, if the value of  $\beta$  increases, the efficiency of the algorithm can be improved and simultaneously solve the problem of pre-mature convergence to some degree.

Regarding the designation and realization of compatibility correction algorithm based on Pearson-Compatibility, readers could refer to [17, 18]. In order to solve the problem of pre-mature convergence, this paper introduces a multi-user collaborative correction mechanism which

is based on Simulated Annealing Algorithm (SAA). The following experiments in this paper testify that this method can solve the problem of pre-mature convergence in advance, and further improves the accuracy of the experiment result. This algorithm is described as follows:

**Step1:** Suppose  $u$  is the iteration times of compatibility test.  $zs$  is the number of users whose evaluation matrix could not be revised further in each step, which is given the original value 0. Suppose original user matrixes of all  $p$  users are  $A^{1(0)}, A^{2(0)} \dots A^{p(0)}$ . Then after  $u$  times compatibility correction iteration,  $p$  users evaluation matrixes are:  $A^{1(u)}, A^{2(u)} \dots A^{p(u)}$

**Step2:** Use formula (2) and (3) to calculate the influence weights of the evaluation matrixes of  $p$  users.

**Step3:** Use formula (4) and (5) to calculate the user comprehensive rating matrix  $X^{(u)}$  after the  $u^{\text{th}}$  compatibility correction.

**Step4:** Use formula (6) to calculate the Compatibility between the  $k^{\text{th}}$  user  $A^{k(u)}$  and  $X^{(u)}$ . Then apply formula (7) to calculate the comprehensive consistency degree of  $\bar{S}^{(u)}$ .

**Step5:** When  $zs = p$  and  $\bar{S}^{(u)} < 0.8$ , go to Step7. When  $zs \neq p$  and  $\bar{S}^{(u)} < 0.8$ , go to Step6. When  $zs \neq p$  and  $\bar{S}^{(u)} \geq 0.8$ , we believe  $p$  users comprehensive consistency could pass the test, go on with Step8.

**Step6:** Referring to [17], we can revise  $X^{(u)}$  and  $A^{k(u)}$ . Suppose the deviation matrix of  $A^{k(u)}$  is  $D^{k(u)} = \{e_{ij}^{k(u)}\}$ , where  $e_{ij}^{k(u)} = x_{ij}^{(u)} - \omega_{ij}^{k(u)}$ . Suppose  $e_{wr}^{k(u)} = \max(|x_{ij}^{(u)} - \omega_{ij}^{k(u)}|)$  is the maximum deviation item after  $u$  times iteration, where  $\omega_{ij}^{k(u)}$  is nonzero term. Let  $u = u + 1$  and begin a new round of revision. The revised rating matrix is:

$$A^{k(u)} = \begin{cases} x_{ij}^{(u-1)} - e_{ij}^{k(u-1)} \times 0.95^\mu, & \text{when } i = w, j = r \\ 1/[x_{ij}^{(u-1)} - e_{ij}^{k(u-1)} \times 0.95^\mu], & \text{when } i = r, j = w \\ x_{ij}^{(u-1)} - e_{ij}^{k(u-1)}, & \text{when } i \neq w, i \neq r, j \neq r, j \neq w \end{cases} \quad (9)$$

where  $u$  is the step length adjusting factor. This paper gives a limitation between 0 and 10 to  $u$ . Thus, the range of  $0.95^\mu$  is  $[0.5987, 1]$ . The larger  $u$  is, the faster the rate of convergence of rating matrix of user  $k$  will be. In order to prevent an oversized adjustment of a single user matrix and thus lose the information in the initial rating matrix, the maximum adjustment extent is set as 0.5987.

If  $u$  is given an appropriate value in each iteration, the calculating speed and the accuracy of the algorithm result will be highly improved. However, the existing algorithm is not effective enough. This paper proposes an approach of building a multi-user collaborative filtering model based on simulated annealing algorithm (SAA), i.e., applying SAA to revise the maximum deviation item in each iteration and calculate the optimal correction factor  $\mu^k$  for each user  $k(k = 1, 2 \dots p)$ .

In the simulated annealing algorithm (SAA), each point  $s$  of the search space is analogous to a state of some physical system. The goal is to bring the system, from an arbitrary initial state, to a state with the minimum possible energy. According to the principle of Metropolis, the probability of solid particle at temperature  $T$  changed from the disordered state to the steady state is  $\exp(-\frac{\Delta E}{k \times T})$ .  $\Delta E$  is the internal energy change quantity when the solid temperature change to  $T$  and  $k$  means the Boltzmann constant. When we apply SAA method to solve the

combinational optimization problem, internal energy  $E$  means the value of target function and temperature  $T$  means the control factor. Then we can get the simulated annealing algorithm.

In the SAA, we suppose the initial energy value of target function in the model is  $E$  and control factor is  $T$ . Then we can control the attenuation amplitude in each iteration and use the random function to simulate the random motion of particle. And we calculate the objective function difference as  $\Delta E$ , and according to Metropolis principle to choose the result. When  $T$  decrease to the critical value or the model solution could not be better, we stop this algorithm and get the approximate optimal result.

The SAA applied to the multi-user collaborative correction algorithm could be described as following:

**Step6.1:** Suppose it is the  $u^{\text{th}}$  iteration. Let formula (8) be the energy function. Suppose  $c = 0$  is the number of iterations for calculating  $\mu^k$ , and  $sc = 0$  is the verdict factor to judge whether to stop the iteration.

**Step6.2:** For user  $A^{k(u)}$ , ( $k = 1, 2, \dots, p$ ), apply formula (9) to confirm the maximal deviation item and to calculate the evaluation set  $A_c^{1(u)}, A_c^{2(u)}, A_c^{3(u)}, \dots, A_c^{p(u)}$ . Randomly choose the initial state set  $[\mu_c^1, \mu_c^2, \dots, \mu_c^p]$  between 0 and 1.

**Step6.3:** Run Matlab software to generate  $p$  random numbers  $\Delta_1, \Delta_2, \dots, \Delta_p$ , which are between 0 and 1. Suppose  $\mu_c^k = \Delta_k$  ( $k = 1, 2, \dots, p$ , and  $\Delta_k$  in each iteration is different). Let  $c = c + 1$  and calculate the energy function  $\Omega_c^{(u)}$ . Suppose  $T$  is the temperature schedule. Its initial value  $T_0 = 100$ . The attenuation function is  $T_c = T_{c-1} \times 0.95$ . Suppose the counting variable is  $rt$  and  $rt = rt + 1$ . Let  $\Delta f = \Omega_c^{(u)} - \Omega_{c-1}^{(u)}$ . Then we make the following judgment:

When  $\Delta f > 0$ ,  $\mu_c^1, \mu_c^2, \dots, \mu_c^p$  is accepted as the new state solution and go on with **Step4**.

When  $\Delta f < 0$ , there are two possibilities. If  $e^{(\frac{\Delta f}{T_c})} > \text{Random}[0, 1]$  holds,  $\mu_c^1, \mu_c^2, \dots, \mu_c^p$  is accepted as the new state solution and go on with **Step4**. Otherwise, let  $\mu_c^k = \mu_{c-1}^k$ ,  $sc = sc + 1$ .

**Step6.4:** When  $sc > 10$  or  $T_c < 0.01$ , which means we reach the end condition and export the result  $\mu_c^1, \mu_c^2, \dots, \mu_c^p$ . Otherwise, go on with **Step2** after the value of  $u$  is obtained.

**Step7:** Popup dialog prompt whether to set a new threshold  $S_{xy}$  which means the user acceptable level. If the user choose to reset,  $zs$  value would be zero. And we will let the value of Pearson in each loop iteration compared with  $S_{xy}$  and get the value of  $zs$ . Then we execute Step5. If the user choose not reset, we execute **Step8**.

**Step8:** Export the calculation result  $X^{(u)}, \bar{S}^{(u)}$  and  $p$  user final evaluation matrixes  $A^{k(u)}$ , ( $k = 1, 2, \dots, p$ ).

We have testify the algorithm is effective in the previous studies. If you are interested in the test process, you can refer to article [16] and [17].

## 4 Experimentation

We build an experiment environment to execute our algorithm at current conditions to validate the effectiveness of this algorithm [6]. The environment is described as follows:

We adopt ontology and the relevant methods in order to design and develop the movie information database. Jena 2.6.2 is applied to store the movie information in RDF format and ARQ-2.2 is used to manage the movie information. We have imported 300 movies which involve 10 categories. A semantic analysis of each movie is conducted to get key words and form the initial attribute set. Then the synonyms and the similar words in the initial set are combined. Take some topical words as the characteristic attributes and use them to represent

these movies. Finally, 15 attributive categories and 282 concrete attributes are extracted. Then an online multi-attribute rating system based on the movie database and a collaborative filtering recommendation system based on group-decision making are designed and developed.

The concrete process that tests the algorithm is as follows [6]:

1. Select four evaluated movies in which  $G(u, p)$  is comparatively big and use them as the testify set. They respectively include 6, 7, 8 and 9 attributes. Then, use the target user evaluation matrixes which are further used as the real weight vectors to calculate the user interest vectors for each movie.

2. Based on the user-evaluated movies set (excluding the 4 movies in the test set), apply the methods in sections 2.3 and 2.4 to searching the most similar user set for the target user (i.e., the similar interest distributions).

Take Movie 1 with 6 attributes as an example. The real interest vectors are  $S = [3.7288, 2.7053, 1.9627, 0.4657, 0.3293, 0.3293]$ . The total score of this movie is 4.5 which indicates that the target user has a high preference to this movie. Moreover, the preference is mainly determined by the first three attributes. Totally, 9 similar users have evaluated this movie. Firstly, the traditional collaborative filtering algorithm is applied to obtaining the weighted average of the total score of this movie and gets the result 3.94. We are not sure whether the target users have interests in this movie. Thus, we need use the similar user evaluation matrixes to make judgments. The evaluation matrixes of six similar users are listed as follows:

$$A = \begin{bmatrix} 1 & 2 & 2 & 5 & 7 & 9 \\ 1/2 & 1 & 1 & 6 & 5 & 6 \\ 1/2 & 1 & 1 & 7 & 7 & 6 \\ 1/5 & 1/6 & 1/7 & 1 & 1 & 2 \\ 1/7 & 1/5 & 1/7 & 1 & 1 & 3 \\ 1/9 & 1/6 & 1/6 & 1/2 & 1/3 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 & 3 & 7 & 7 \\ 1/2 & 1 & 0 & 4 & 7 & 6 \\ 1/3 & 0 & 1 & 0 & 4 & 5 \\ 1/3 & 1/4 & 0 & 1 & 1 & 2 \\ 1/7 & 1/7 & 1/4 & 1 & 1 & 3 \\ 1/7 & 1/6 & 1/5 & 1/2 & 1/3 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 2 & 4 & 0 & 0 \\ 1/2 & 1 & 1 & 3 & 0 & 0 \\ 1/2 & 1 & 1 & 2 & 0 & 0 \\ 1/4 & 1/3 & 1/2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 2 & 3 & 4 & 6 & 8 \\ 1/2 & 1 & 2 & 2 & 4 & 6 \\ 1/3 & 1/2 & 1 & 1/2 & 4 & 7 \\ 1/4 & 1/2 & 2 & 1 & 3 & 2 \\ 1/6 & 1/4 & 1/4 & 1/3 & 1 & 2 \\ 1/8 & 1/6 & 1/7 & 1/2 & 1/2 & 1 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 2 & 0 & 7 & 8 & 0 \\ 1/2 & 1 & 2 & 8 & 7 & 7 \\ 0 & 1/2 & 1 & 0 & 7 & 8 \\ 1/7 & 1 & 0 & 1 & 1 & 2 \\ 1/8 & 1/7 & 1/7 & 1 & 1 & 1 \\ 0 & 1/7 & 1/8 & 1/2 & 1 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 2 & 4 & 0 & 7 & 0 \\ 1/2 & 1 & 2 & 6 & 7 & 7 \\ 1/4 & 1/2 & 1 & 0 & 2 & 2 \\ 0 & 1/6 & 0 & 1 & 0 & 2 \\ 1/7 & 1/7 & 1/2 & 0 & 1 & 1 \\ 0 & 1/7 & 1/2 & 1/2 & 1 & 1 \end{bmatrix}$$

3. Apply the four algorithms to calculate the score of the four movies and make comparisons on the deviations of the real weight vectors of the target users. The result is listed as follows:

Table 1: The comparison between algorithms

	Movie 1 (6 order)	Movie 2 (7 order)	Movie 3 (8 order)	Movie 4 (9 order)
Arithmetic weighted average method	0.1589	0.0564	0.1985	0.1132
Logarithmic least squares method	0.1054	0.0534	0.1398	0.0831
Compatibility correction algorithm	0.0877	0.0556	0.1042	0.0687
Our algorithm	0.0780	0.0543	0.0885	0.0683

When the scores of a part of similar users have a large deviation from those of the other users, the algorithm proposed in this paper can solve the problem of early convergence better than the other algorithms and obtain an accurate result, as shown in Table 1. The core of our algorithm is the revised values of the comprehensive evaluation matrix determined by the majority of users. Accordingly, the highly deviated evaluation values are revised. The result of seven order matrix experiment shows that the deviations of the result of any algorithms are not notable when all the similar users have unanimous evaluation matrixes, The result of nine order matrix experiment shows that the result of the proposed algorithm is similar to that of compatibility correction algorithm when all the similar users have unanimous evaluation matrixes, while still have some incomplete values, and is better than the other two algorithms obviously. When there are 5 similar users and six order evaluation matrix is executed with our algorithm, the change tendencies of the main indicators are shown in Figure 1:

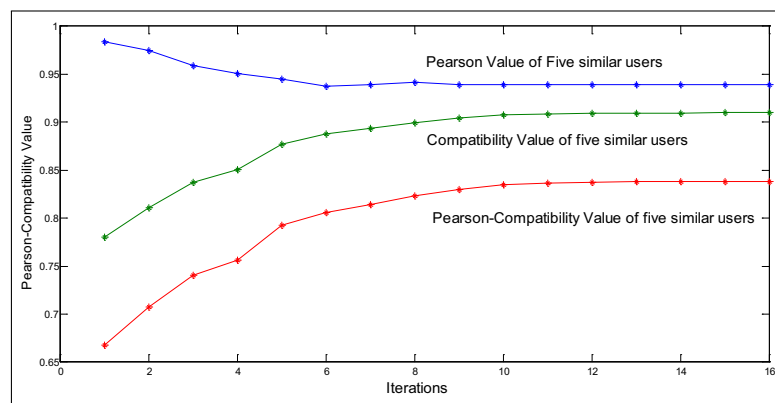


Figure 1: The main indicators change in our example

4. For example, for the first movie with the six order evaluation matrix, the influence of the user number on the accuracy of recommendation results is examined. Suppose the user number is 3, 5, 7 and 9. The accuracy and the number of iterations are calculated with different means of permutation and combination. Part of the result is shown in Table 2:

Table 2: The comparison of different similar user numbers

User number	3	5	7	9
initial Indicator of Comprehensive consistency degree	0.5872	0.6890	0.6872	0.7081
Deviation of result	0.1680	0.1093	0.0828	0.0780
Number of iterations	12	16	28	37

More deviation items are generated as the user number increases. Therefore, the iterations of this algorithm rise. This test indicates that the effectiveness of this algorithm is highly related to the initial consistency degrees of all users and the number of users. In general, when the initial consistency degree is low and the similar user set is limited (e.g. there are 3 users), it is hard for the algorithm to dig out the common information among the users. Therefore, the result deviation is huge. However, when the number of similar users increases to a certain degree (e.g. the number is equal or bigger than 7), the algorithm still remains a good accuracy, even if the initial compatibility is low.

Regarding the provision of personalized services to the target users, this paper calculates the comprehensive evaluation weight vectors of each movie with the group-decision making model.

Take the first movie with 6 attributes as an example. The comprehensive evaluation score of nine similar users is  $G(u, p) = 3.94$ . The comprehensive evaluation vectors are  $V = [4.0653 \ 2.9492 \ 1.7630 \ 0.3972 \ 0.3044 \ 0.3607]$ . Each value of the weight vector represents the potential interest degree of the target user on the corresponding product attributes. Thus, the total score calculation formula is

$$TScore = G(u, p) \times \sum_{i=1}^n V_i/n \quad (10)$$

where  $TScore$  denotes the total score,  $n$  denotes the number of attributes, and  $V_i$  denotes the comprehensive evaluation value of the  $(i)$ th attribute of the product. The recommendation set can be fixed through the way of ranking or threshold setting. The total scores of the four movies is shown in Figure 2:

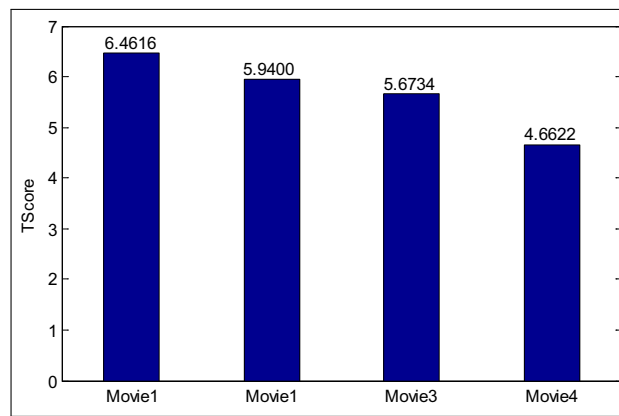


Figure 2: The TScore of four movies

The first movie has the highest score with six attributes, as shown in Figure 2. The characters of this movie are analyzed as follows using the user interest model. Firstly, attribute  $V_i$  of weight vector  $V$  is normalized and generate vector  $\bar{V} = [0.4131, 0.2997, 0.1792, 0.0404, 0.0309, 0.0367]$ . The three attributes whose values are bigger than the average value 0.1666 are picked out. When the attribute value is bigger than 0.1666, the majority of users have evident preference on movie one. In the target user interest model, there are 124 attributes totally. The 3 attributes of movie one that are bigger than the average value are connotation, characteristic and special efficiency are still larger than the average value ( $1/124 = 0.0081$ ) among 124 target user attribute preferences. This result indicates that target user has evident preference to these 3 attributes and the popularity of this movie is mainly determined by these attributes. Therefore, we could introduce movie one to target user and provide the reasons why this movie is introduced. We also could use semantic analysis technique to describe each attribute in detail and provide more personal service to target user. The comparison analysis histogram is shown in Figure 3.

M bar means the percentages which attributes are ranking at top 3 in movie one, U bar means percentages which attributes are ranking at top 3 in target user interest model. The analysis show us that target user may have larger interest preference to attribute one in movie one.

## 5 Conclusion

The traditional collaborative filtering personal recommended algorithms seldom consider the multi-attribute problem. Our approach is based on group-decision making. We propose an

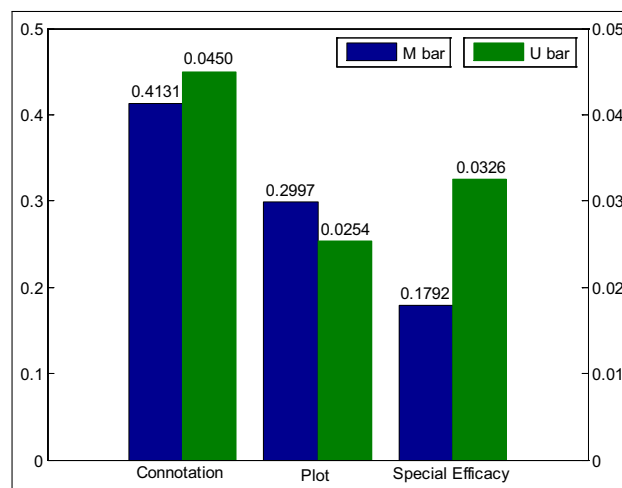


Figure 3: Comparison analysis histogram of movie one with target user interest.

improved Pearson-Compatibility algorithm which is applied to the collaborative filtering recommend field. We then build a virtual recommend environment and testify the effectiveness and feasibility of this algorithm. The advantages of the collaborative filtering personal recommended algorithm based on group-decision include:

Identifying a more suited similar users set for the target user. An accurate target user model could be set up via field subdivision according to field attributes. Then the users who have similar interest distribution with target user can be found. The similar user set is generated.

Providing more accurate and personal recommend service to the target user. The traditional collaborative filtering method could neither recommend a result set to target user, nor provide analysis service [6]. This weakness is overcome by making an information integration to know what are mainly factors determining the user preference, so that we could handle the user need more accurate.

Considering evaluation deviation between the similar users and revising the user evaluation. Instead of weighted mean, group-decision making method is used to calculate the comprehensive evaluation score. Deleting the deviation item and revising the evaluation matrix could make the result have a better fitting effect [6]. The collaborative filtering method based on Pearson-Compatibility is applied to the personal recommended field. The result of the experiment shows that the algorithm is stable when dealing with the deviation items and identifies the common preference information between similar users.

## Acknowledgment

This research work is supported by Innovation Program of Shanghai Municipal Education Commission (No. 12ZZ070, No. 12YS018) and Humanity and Social Science Youth Foundation of Ministry of Education, China (No. 12YJC630136).

## Bibliography

- [1] G. Adomavicius, A. Tuzhilin (2005); Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734-749.

- [2] Xiaoyuan Su, Taghi M. Khoshgoftaar(2009); A survey of collaborative filtering techniques, *Advances in Artificial Intelligence*, 2009:1-19.
- [3] D. Goldberg, D. Nichols, B. M. Oki, D. Terry(1992); Using collaborative filtering to weave an information tapestry, *Communications of ACM*, 35(12):61-70.
- [4] P. Resnick, H. R. Varian(1997); Recommender systems, *Communications of the ACM*, 40(3):56-58.
- [5] K. Goldberg, T. Roeder, D. Gupta, C. Perkins (2001); Eigentaste: a constant time collaborative filtering algorithm, *Information Retrieval*, 4(2):133-151.
- [6] C. Yu, Y. Luo, K. Liu(2014); A multi-attribute collaborative filtering recommendation algorithm based on improved group decision-making, *ICISO 2014, IFIP AICT*, 426:320-330.
- [7] F. Herrera, E. Herrera-Viedma, F. Chiclana (2001); Multiperson decision-making based in multiplicative preference relation, *European J. of Operational Research*, 129:372-385.
- [8] A. Shepitsen, J. Gemmell, B. Mobasher, R. Burke (2008); Personalized recommendation in social tagging systems using hierarchical clustering, *Proc. of the 2008 ACM conference on Recommender systems*, Lausanne, Switzerland.
- [9] A. Ypma, T. Heskes (2002); Categorization of Web pages and user clustering with mixtures of hidden Markov models, *Proc. of the WEBKDD 2002 Workshop: Web Mining for Usage Patterns and User Profiles*, SIGKDD 2002, Edmonton, Alberta, Canada.
- [10] Z.S. Hua, B.G. Gong, X.Y. Xu (2008); A DS-AHP approach for multi-attribute decision making problem with incomplete information, *Expert Systems with Applications*, 34(3):2221-2227.
- [11] H. Jeon, T. Kim, J. Choi (2010); Personalized information retrieval by using adaptive user profiling and collaborative filtering, *AISS: Advances in Information Sciences and Service Sciences*, 2(4):134-142.
- [12] Liu, A., Yang, Z.(2010); Watching, thinking, reacting: a human-centered framework for movie content analysis, *International Journal of Digital Content Technology and its Applications*, 4(5):23-37.
- [13] Y. Dong , Y. Chen, S. Wang (2004); Algorithm of solving weights for group decision making by improving compatibility, *Systems Engineering-theory & Practice*, 2004-10.
- [14] C.-S. Yu (2002); A GP-AHP method for solving group decision-making fuzzy AHP problems, *Computers & Operations Research*, 29(14): 1969-2001.
- [15] Y. Zhang, H. Wang (2002); Development and application of P-S aided decision system, *Systems Engineering-Theory Methodology Application*, 2002-04.
- [16] L. Liang, L. Xiong, G. Wang (2004); A new method of determining the reliability of decision-makers in group decision, *Systems Engineering*, 22(6):91-94.
- [17] J. Barzilai, F. A. Lootsma (1994); Power relations and group aggregation in the multiplicative AHP and SMART, *Proc. 3rd Int. Symp. AHP*, 157-168.
- [18] J. Sun, W. Xu, Q.D. Wu(2005); A new algorithm for incomplete matrixes' compatibility improvement and group ranking in group decision making, *Systems Engineering-Theory & Practice*, 10(10):89-94.